面向移动云计算任务调度的改进鸟群算法研究 *

陈 暄 [1]，赵文君 [2]，龙 丹 [3]

(1. 浙江工业职业技术学院, 浙江 绍兴 312000; 2. 北京信息科技大学, 北京 100101; 3. 浙江大学, 杭州 310058)

摘 要：针对移动云计算环境下的任务调度存在耗时长，设备能耗高的问题，提出了一种改进的鸟群算法(improved bird swarm algorithm，IBSA)的任务调度策略。首先，构建了以能耗和时间为主的移动云任务调度模型，其次，提出了自适应感知系数和社会系数，避免了算法陷入局部最优；构建了学习因子优化飞行行为，保证了个体寻优能力；最后，任务调度目标函数作为鸟群个体的适应度函数参与算法的迭代更新。仿真结果表明该算法相比于蚁群算法，粒子群算法，鲸鱼算法，鸟群算法在移动云计算任务调度方面具有良好的效果，能够有效的节省时间和降低能耗。

关键词：移动云计算；鸟群算法；自适应；学习因子

中图分类号：TP393 doi: 10.19734/j.issn.1001-3695.2020.02.0043

Research on task scheduling of improved bird group algorithm for mobile cloud computing

Chen Xuan[1], Zhao Wenjun[2], Long Dan[3]

(1. *Zhejing Industry Polytechnic College*, *Zhejing Shaoxing* 312000, *China*; 2. *Beijing Information Science & Technology University*, *Beijing* 100101, *China*; 3. *Zhejiang University*, *Hangzhou* 310058, *China*)

Abstract: Aiming at the problem of long time-consuming and high equipment energy consumption for task scheduling in mobile cloud computing environment, a task scheduling strategy based on improved bird group algorithm is proposed, a task scheduling strategy based on Improved Bird Swarm Algorithm (IBSA) is proposed. Firstly, a mobile cloud task scheduling model based on energy consumption and time is constructed. Secondly, adaptive sensing coefficients and social coefficients are proposed to prevent the algorithm from falling into a local optimum. Learning factors are optimized to optimize flight behavior and ensure that Superior ability. Finally, . the task scheduling objective function is used as the fitness function of the bird group to participate in the iterative updating of the algorithm. The simulation results show that the algorithm has good effects in mobile cloud computing task scheduling compared with ant colony algorithm, particle swarm algorithm, whale algorithm and bird swarm algorithm, which can effectively save time and reduce energy consumption.

Key words: mobile cloud computing; bird swarm algorithm; adaptive; learning factor

0 引言

无线移动计算方式借助便携式设备帮助人们跨地域是进行办公、协助企业完成资源调配等工作，但是受限于电池容量、存储和计算能力方面的约束，该方式无法与传统服务器和桌面设备对比，移动设备的硬件能力非常有限[1]。移动云计算技术(mobile cloud computing)[2~4]的出现提供了一种良好解决思路。与普通的云计算不同，移动云计算中节点由大量移动终端组成，这些终端设备的自身存储容量，网络连接状态，设备电量等属性影响着整个云任务的调度。文献[5~7]和文献[8~10]仅仅从任务完成时间或者设备能耗方面进行单一研究。部分学者将其进行统一优化考虑，如文献[11]提出了多任务流下的 CCS 算法，通过增加任务之间传输与执行的并发性，增加任务集整合的概率，提高任务处理的速率，减少任务的响应时间，但是缺乏与其他的调度算法对比；文献[12]考虑了执行时间联合优化的任务调度模型和目标方程。使用模拟退火算法的任务调度，缩短了时间，降低了能耗，但是提高了算法的复杂度；文献[13]在移动云计算的任务调度采用一种任务迁移的线性时间调度算法，算法的实践效果有待进一步验证；文献[14]提出了基于鲸鱼算法的最优任务工作流调度方案，但是增加了算法运行时间；文献[15]提出了一种基于蚁群优化算法的协同多任务调度方案，但缺乏与较新的仿生学算法进行对比。文献[16]采用了仿生学算法解决用户移动设备的任务调度问题，算法在能效和响应时间方面具有良好的调度效果。上述研究表明采用仿生学的算法能够提高移动云计算下的任务调度效果。受此启发，本文将优化后的鸟群算法用于移动云计算任务调度中。对任务耗时和设备能耗展开研究。

1 相关知识

1.1 移动云计算任务调度

移动云计算环境下的任务分配目标是将 N 个可并行化处理的子任务合理地分配给 M 个移动设备执行，尽可能的减少任务执行时间和系统电量消耗，提高任务执行效率，降低系统总能耗。本文以文献[17]中阐述的任务时间和设备能耗计算方式为基础，分别构建了任务时间适应度函数和设备能耗适应度函数。将用户提交的任务划分成为 N 个能够并行处理的子任务因此设 $T=\{T_1,T_2\cdots T_N\}$ 表示任务集合。

a) 任务完成时间适应度函数

移动云计算环境中执行任务的移动资源具有动态性和异构性，同样的任务在不同的移动设备上完成时间可能是不相同的，因为这与设备自身的计算能力有关，通常来说，云服务器分配给某一个设备 D_j 的任务 T_i，它的执行时间 $te_{i,j}$ 取决于任务 T_i 自身的长度 L_i 和该设备的 CPU 具有的处理能力 C_j (MIPS)，因此执行时间表达式为

$$te_{i,j}=\frac{L_i}{C_j\times(1-u_{cpu})} \tag{1}$$

收稿日期：2020-02-22；**修回日期：**2020-04-08 **基金项目：**国家自然科学基金资助项目(LQ18A010003,11426205)

作者简介：陈暄(1979-)，男，江西人，副教授，硕士，主要研究方向为云计算、无线传感(chenxuan1979@sina.com)；赵文君(1996-)男，山西人，硕士研究生，主要研究方向为 5G 网络、云计算、算法设计；龙丹(1975-)，男，湖南人，讲师，博士，主要研究方向为图像处理和分析、算法设计.

其中，u_{cpu} 表示移动设备 CPU 使用率，需要说明的是，考虑 u_{cpu} 主要是基于移动设备中的 CPU 在无论是在空闲状态还是忙碌状态所消耗的时间必定会影响任务的整体执行时间。

每一个移动云任务的完成时间除了和设备的计算能力有关外，通常与该设备所处在网络中的传输能力有关，这是由于不同的移动设备的网络带宽不尽相同，它们对任务映射到该设备的时间以及从设备返回运行结果的时间产生影响，进而影响任务完成整体时间。设定对于分配给移动设备 D_j 的任务 T_i，任务映射时间和返回时间分别是 ts_{ij} 和 tr_{ij}，主要是由输入数据大小 d_i^{in} 和 d_i^{out} 以及设备的网络带宽 B_j 决定，因此分别为

$$ts_{ij} = \frac{d_i^{in}}{B_j} \tag{2}$$

$$tr_{ij} = \frac{d_i^{out}}{B_j} \tag{3}$$

因此某个移动设备 D_j 完成任务 T_i 的时间 t_{ij} 为

$$t_{ij} = te_{ij} + ts_{ij} + tr_{ij} \tag{4}$$

任务 T_i 的最终完成时间为耗时最长的子任务所需的时间，即 t_T 表达式如下：

$$t_T = \max_{ij} t_{ij} \tag{5}$$

因此该任务完成的时间适应度函数 fit_t 为

$$fit_t = \min_{ij} t_T \tag{6}$$

b) 设备能耗适应度函数

移动云计算环境中，移动设备的有限电量是不可忽视的约束条件。移动设备属于电量有限的计算资源，在任务分配的过程中除了考虑任务执行中所产生的电量能耗，还必须考虑设备自身产生的电量能耗，所以每一个移动设备的电量首先要保证自身必要控制模块程序的正常运行，其次剩余的电量用于移动云计算下的调度任务。根据文献[18]中描述的移动设备的 CPU，内存，WiFi 等硬件模块的能耗，因此移动设备的能耗模型为

$$p_{ij} = \beta_{ij} \times c_j \tag{7}$$

其中，p_{ij} 表示测试样本中的第 i 个样本的第 j 个硬件设备产生的功耗，β_{ij} 表示第 i 个样本涉及的第 j 个硬件设备的使用率(CPU 使用率或者内存使用率)，c_j 表示第 j 个硬件设备使用率的相关耗电系数。计算机的能耗与 CPU 的利用率是线性关系[19]，表达式为

$$E_{cpu} = \alpha_{cpu} \times \mu_{cpu} + \gamma_{cpu} \tag{8}$$

其中，E_{cpu} 表示 CPU 能耗，μ_{cpu} 表示使用率，α_{cpu} 和 γ_{cpu} 分别表示固定系数。

借助以上的模型以及移动云计算中终端的特点，消耗电量主要来自 CPU 和内存模块，因此移动设备 D_j 执行任务 T_i 的产生的能耗如下：

$$er_{ij} = \mu_{cpu} \times c_{cpu} \times te_{ij} + u_{mem} \times c_{mem} \times te_{ij} \tag{9}$$

其中，μ_{cpu} 和 u_{mem} 分别表示 CPU 使用率和内存使用率，c_{cpu} 和 c_{mem} 分别表示 CPU 和内存模块的电量消耗系数，te_{ij} 为任务 T_i 在设备 D_j 上执行时间。

除了以上的能耗产生之外，当数据在每一个移动设备与各个代理服务器之间进行传输的时候也会产生部分能耗，主要是由于不同的移动设备采用不同的网络连接方式造成了设备的功耗也不相同。根据文献[20]的描述移动设备的数据传输能耗与传输数据大小之间的正比关系，设定本文的移动设备数据传输引起的能耗 ed_{ij} 如下：

$$ed_{ij} = d_i^{in} \times ts_{ij} \times c_n + d_i^{out} \times tr_{ij} \times c_n \tag{10}$$

其中，d_i^{in} 和 d_i^{out} 分别表示任务 T_i 的输入数据大小和输出数据大小，c_n 为网络传输模块的功耗系数，ts_{ij} 为任务映射时间，tr_{ij} 为结果返回时间。

移动资源设备 D_j 完成任务 T_i 的能耗 e_{ij} 为

$$e_{ij} = er_{ij} + ed_{ij} \tag{11}$$

任务 T_i 最终完成能耗为所有任务能耗总和即 e_T 表达式如下：

$$e_T = \sum_{ij} e_{ij} \tag{12}$$

因此，设备能耗的适应函数如下：

$$fit_e = \min_{ij} e_T \tag{13}$$

1.2 鸟群算法

BSA 算法[21]是 2015 年由 Xian-Bing Meng 提出的一种基于新的群体智能算法。该算法灵感来源于对鸟群行为的研究，通过模仿鸟群捕食中出现的觅食行为、警戒行为和飞行行为而构建的一种模型，它通过基于信息共享机制及搜索策略而获得最优解，在解决最优化问题的时候的性能明显优于蚁群算法，粒子群算法。设定鸟群种群规模为 N，搜索空间为 D 维，第 i 只鸟在第 D 维空间中的位置表示为 $X_i = (X_i^1, X_i^2, \cdots X_i^D)$，$i \in [1, 2, \cdots N]$。

a) 觅食行为

每一只鸟个体在进行觅食的时候通常借助自身和整个种群的经验进行觅食，个体位置更新如下

$$x_{i,j}^{t+1} = x_{i,j}^t + (p_{i,j} - x_{i,j}^t) \times C \times rand + (g_j - x_{i,j}^t) \times S \times rand \tag{14}$$

其中，$x_{i,j}^{t+1}$ 和 $x_{i,j}^t$ 分别表示第 i 只鸟在第 j 维中的 $t+1$ 次和 t 次迭代中个体所在的位置，$rand$ 表示(0,1)之间的随机数，C 和 S 分别为感知系数和社会驾驶系数，$p_{i,j}$ 表示第 i 只鸟在第 j 维中的最佳位置，g_j 表示整个鸟群群体在第 j 维所处的最佳位置。

b) 警戒行为

鸟群在飞行的过程中都会向着种群中心移动，这个时候就会不可避免的产生个体之间的竞争关系，从而造成飞行中的阻碍。因此，警戒行为下的个体位置更新公式如下：

$$x_{i,j}^{t+1} = x_{i,j}^t + A_1 \times (mean_j - x_{i,j}^t) \times rand + A_2 \times (p_{k,j} - x_{i,j}^t) \times rand \tag{15}$$

$$A_1 = a_1 \times \exp(-\frac{pFit_i}{sumFit + \varepsilon} \times N) \tag{16}$$

$$A_2 = a_2 \times \exp((\frac{pFit_i - pFit_k}{|pFit_k - pFit_i| + \varepsilon}) \frac{N \times pFit_k}{sumFit + \varepsilon}) \tag{17}$$

其中，$mean_j$ 表示在第 j 维中的平均位置，$p_{k,j}$ 表示第 k 只鸟在第 j 维中的位置，k 为 $[1, N]$ 之间的随机正整数且 $k \neq i$。a_1 和 a_2 为[0,2]之间的常数，$pFit_i$ 和 $pFit_k$ 分别表示第 i 只鸟和第 k 只鸟的最佳适应度值，$sumFit$ 表示整个种群的最佳适应值总和，ε 为分母避免为 0 的一个很小的常量。A_1 和 A_2 分别表示鸟群个体飞向种群中心的过程中由于周围环境引起的间接影响和直接影响。当 Fit_k 优于 Fit_i，说明个体 i 受到的干扰大于个体 k，因此个体 k 也可能向着种群中心移动。

c) 飞行行为

鸟群在飞行过程中，可能会受到来自外界的其他飞行动物的干扰而飞行到其他地方重新开始生活，当在新的地方开始寻找食物的时候，鸟群中一些鸟个体扮演生产者身份寻找食物，而剩余的鸟群个体可能会扮演乞食者的身份跟随生产者。因此生成者和乞食者分别定义如下

$$x_{i,j}^{t+1} = x_{i,j}^t + randn(0,1) \times x_{i,j}^t \tag{18}$$

$$x_{i,j}^{t+1} = x_{i,j}^t + (x_{k,j}^t - x_{i,j}^t) \times FL \times rand \tag{19}$$

其中，$randn(0,1)$ 表示均值为 0，标准偏差为 1 的高斯分布，$k \in [1, N]$，$k \neq i$，$FL \in (0, 2]$ 表示乞食者跟随生产者寻找食物，鸟群飞到其他地方的频率 FQ 为个正整数。

2 改进的鸟群算法的移动云计算任务调度

针对 BSA 算法存在易陷入局部最优和过早收敛的缺点，

文献[22]通过引进种群相似度和聚集度概念、随机概率赋值寻优位置的等方法改善算法后期易陷入局部最优的状况;文献[23]采用当前最优个体替代原算法中随机选取的方法以及步长加权平均思想提高了算法的性能;文献[24]采用了迁移策略和变异策略提高鸟群的局部寻优能力。以上改进的措施在不同程度上提升 BSA 算法的性能,但存在算法复杂度高,算法收敛效果不明显的缺点。

2.1　改进的鸟群算法

a) 自适应感知系数和社会系数

在觅食行为中，每一个个体都需要借助自身的位置和种群的位置完成更新，本文针对感知系数 C 和社会驾驶系数 S 进行优化，通过设置不同的系数值保证每个鸟群个体获得不一样的飞行效果。新的感知系数 C_1 和社会驾驶系数 S_1 如下：

$$C_1 = C_{\min} + C_{\min} \times \cos(\frac{t}{t_{\max}}) \times \frac{1}{f_{\max}(x_i^t) - f_{\min}(x_i^t) + 1}) \tag{20}$$

$$S_1 = 2 - C_1 \tag{21}$$

其中，$C_{\min}$ 为感知系数最小值,t 为当前迭代次数,$t_{\max}$ 为最大迭代次数，从式(20)中发现在算法的初期，个体的 $\frac{1}{f_{\max}(x_i^t) - f_{\min}(x_i^t) + 1}$ 数值较小，C_1 的数值较小，算法侧重全局最优搜索,但是随着迭代次数增加，$\frac{1}{f_{\max}(x_i^t) - f_{\min}(x_i^t) + 1}$ 数值逐渐变大，C_1 值逐渐变大,方便后期进行精细搜索，提高了算法收敛精度，保证鸟群个体以便跳出局部最优，降低了鸟群个体错过最优解的概率。

b) 基于学习因子的飞行行为

在飞行行为中，鸟群中的生产者会继续觅食，而乞食者以概率方式从设定生产者那里获得食物，这样的方式没有考虑到乞食者个体在更新的时候与子群中的其他乞食者的位置信息，可能出现乞食者会随着生产者一起陷入局部最优的情况。因此，本文对乞食者位置公式进行优化，加入了当前乞食者向其他乞食者进行学习的因素，优化后的乞食者位置公式如下。

$$x_{i,j}^{t+1} = x_{i,j}^t + (x_{k,j}^t - x_{i,j}^t) \times FL \times rand \times M \tag{22}$$

其中：

$$M = \exp(\frac{f_i^{'}(x_i^t)}{f_g^{'}}) \tag{23}$$

其中,M 表示学习因子，$f_i^{'}(x_i^t)$ 表示当前乞食者个体适应度函数，$f_g^{'}$ 当前子群中的最优乞食者个体适应度。$\frac{f_i^{'}(x_i^t)}{f_g^{'}}$ 比值表明了第 i 个乞食者与子群内的最优乞食者之间的关联程度。当比值越大，则 M 值越大,说明当前乞食者个体越接近最优乞食者，进一步保证了乞食者个体的寻优能力，为后续的飞行提供了保障。

2.2　适应度函数

本文将完成时间适应度函数和能耗适应度函数作为任务目标调度函数,将鸟群个体对应每一个任务调度方案,任务目标调度函数作为鸟群算法的适应度函数,依靠鸟群算法自身的良好性能,通过适应度函数比较获得最优的鸟群个体,即获得最优的调度方案。但由于时间适应度函数和能耗适应度函数两者取值范围差异较大,采用归一化的方法进行调整,使得调整后的函数值在[0,1]范围内。设 $f(x)$ 分别表示 e_T 和 t_T，$f_{\max}(x)$ 和 $f_{\min}(x)$ 分别表示函数 $f(x)$ 的最大值和最小值,调整后的适应度函数 $f^{'}(x)$ 分别表示 $e_T^{'}$ 和 $t_T^{'}$,表达如下:

$$f^{'}(x) = \frac{f(x) - f_{\min}(x)}{f_{\max}(x) - f_{\min}(x)} \tag{24}$$

因此，鸟群算法的适应度函数为调整后的时间和能耗函数和

$$fit_T = \alpha \times t_T^{'} + \beta \times e_T^{'} \tag{25}$$

其中,$t_T^{'}$ 和 $e_T^{'}$ 分别为调整后的任务完成时间函数和系统设备能耗函数。α 和 β 分别为权重且 $\alpha + \beta = 1$,因此求出 $\min(fit_T)$ 是鸟群算法的目标。

2.3　调度流程

a) 初始化鸟群算法及其他相关参数,将移动云计算任务调度方案与鸟群个体进行一一对应,设定最大的迭代次数;

b) 对鸟群算法的觅食行为进行优化;

c) 对鸟群算法的飞行行为进行优化;

d) 将任务目标函数值和上一次迭代中的目标函数值进行对比,如果小于上一次迭代的结果则取代原来鸟群个体位置,否则保持不变。

e) 当迭代次数小于最大迭代次数,转步骤c),否则转步骤f);

f) 输出最优适应度鸟群个体位置,即为最佳的云计算任务方案。

3　仿真实验

3.1　实验环境及参数

本实验选择硬件 CPU 主频为 2.4 GHz，8G DDR3，1 TB 硬盘，操作系统为 Windows 7，仿真环境为 MATLAB2012B，运行环境为 Java Hotspot 64bit Server VM。实验中的主要参数如表 1 所示。相关对比算法的参数如表 2 所示。

表 1　任务分配相关参数

Tab. 1　Task allocation related parameters

参数		取值
移动设备	CPU 处理能力(MIPS)	3000-10000[间隔为 1000]
	网络传输能力(bit/s)	100-1000[间隔为 100]
	CPU 使用率	0.1-0.9[间隔为 0.1]
任务	任务长度(百万指令)	10000-20000[间隔为 1000]
	传输数据	100-1000[间隔为 100]

表 2　对比算法参数

Tab. 2　Compare algorithm parameters

算法名称	说明
蚁群算法(ACO)	信息素值为 0.005,信息素挥发系数为 0.01,路径选择概率为 0.5
粒子群算法(PSO)	惯性权重为 0.5,两个学习因子为 0.5,随机数权重设为 0.5
鲸鱼算法(WOA)	递减系数 a 在[2,0]中线性递减
鸟群算法(BSA)	感知系数和社会驾驶系数都为 1.5, a_1 和 a_2 为 1, $P \in [0.8,1]$, $FL \in [0.5,0.9]$, FQ 为 3.
改进的鸟群算法 ABSA[22]	感知系数和社会驾驶系数都为 1.5, a_1 和 a_2 为 1, $P \in [0.8,1]$, $FL \in [0.5,0.9]$, FQ 为 10, $\tau \in [0.5,0.9]$, $\lambda = 6.8$, $a = 0.3$, $b = 0.1$, $\delta_{\max} = 1$, $\delta_{\min} = 0.1$
本文算法(IBSA)	感知系数最小值为 1, a_1 和 a_2 为 1, $P \in [0.8,1]$, $FL \in [0.5,0.9]$, FQ 为 3, $rand$ 为 0.5,初始尺度系数 a_{init} 为 1

3.2　实验内容

本次实验内容分为三个部分:一是任务调度函数评价;二是测试本文算法不同任务数量下的任务完成时间和移动设备能耗的情况;三是将本文算法对比其他算法在不同任务数量下的完成时间和设备能耗情况。设定移动设备数量为[50,500],任务数量为 100、200、500 和 1000 的四种条件。

a) 任务调度函数评价

图 1 显示了六种算法在移动云计算下的任务调度评价函数值的对比结果。本文提出的 IBSA 算法相比于其他五种算法能够快速达到最小的稳定值,同时也反映 IBSA 算法相比于 BSA 算法具有良好的收敛性能,因此说明了 IBSA 算法在移动云计算下任务中具有较好的调度效果。

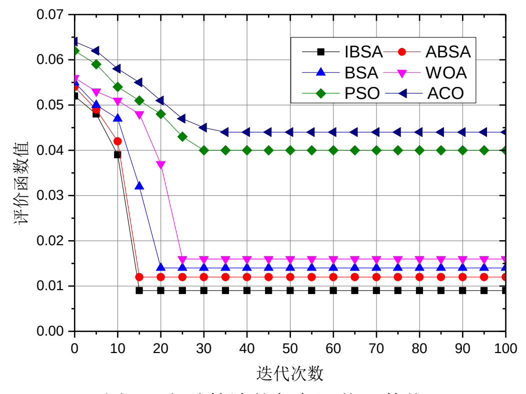


图 1　六种算法的任务评价函数值

Fig. 1　Comparison of task evaluation function values of six algorithms

b) 本文算法的任务完成时间和设备能耗对比

图 2、3 显示本文算法在任务调度完成时间和能量消耗的对比情况。图 2 显示当任务数量不变的时候，任务完成时间随着移动设备数量的不断增加而逐步减小。当任务数量较大的时候，通过增加移动设备数量能够明显有效的降低任务完成时间，当任务数量较小的时候，任务完成时间下降趋势变缓慢，这表明任务数量较小，增加移动设备数量并无法明显的加快任务完成速度；图 3 显示移动设备数量一定时候，任务完成能耗随着移动设备数量不断增加而增加，当任务数量越多则设备能耗越高。在任务数量不变的情况下，移动设备能耗几乎不会随移动设备数量的增加而大幅度增长，这表明对于数量不变的任务，云服务器调度完成这些任务所要需要的设备能耗也是固定的。

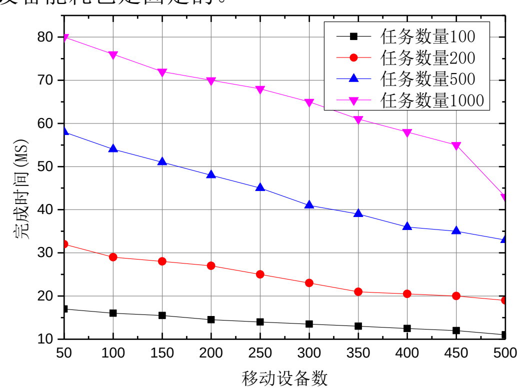


图 2　不同移动设备数量的四种任务完成时间对比

Fig. 2　Comparison of the time to complete four tasks by different numbers of mobile devices

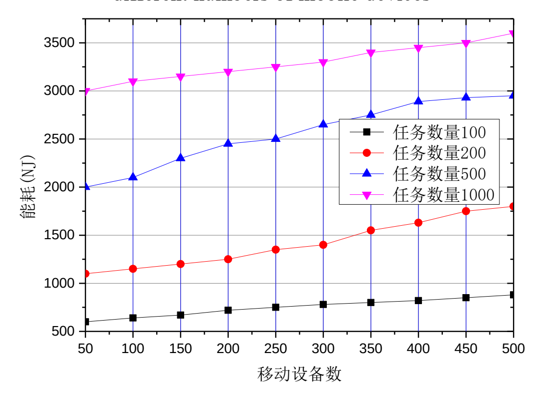


图 3　不同移动设备数量的四种任务数量能耗对比

Fig. 3　Comparison of four task energy consumption by different numbers of mobile devices

c) 与其他算法在任务完成时间和设备能耗对比

图 4、5 显示了六种算法在任务数量为 100 的完成时间和能耗对比结果,六种算法在完成时间和能耗上对比相差不大,这说明任务数量较小的情况下本文算法并不占据优势。图 6、7 显示了六种算法在任务数量为 1000 的完成时间和能耗对比结果,在移动设备数逐渐增多的情况下,本文算法的任务完成时间远低于其他五种算法，说明任务完成的时间上具有一定优势,得益于算法性能得的提升。在能量消耗中,本文算法的系统设备能耗最小,其次为 ABSA 算法、WOA 算法、PSO 算法,而 ACO 算法的能耗最高,这说明本文算法能够最大限度的节约设备能耗。

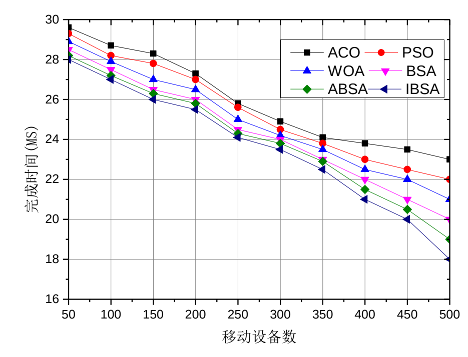


图 4　任务数量 100 的六种算法完成时间对比

Fig. 4　Comparison of six Algorithms' completion time with 100 tasks

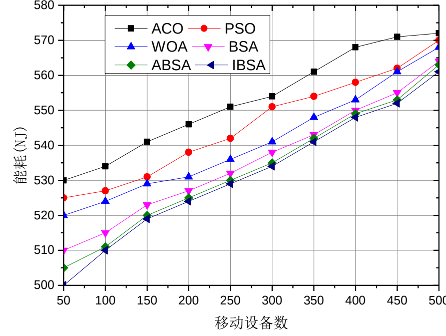


图 5　任务数量 100 的六种算法设备能耗对比

Fig. 5　Comparison of six Devices' energy consumption with 100 tasks

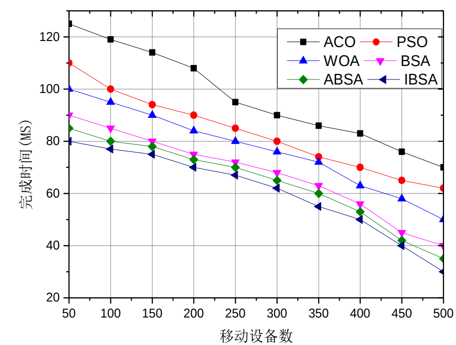


图 6　任务数量 1000 的六种算法完成时间对比

Fig. 6　Comparison of six Algorithms' completion time with 1000 tasks

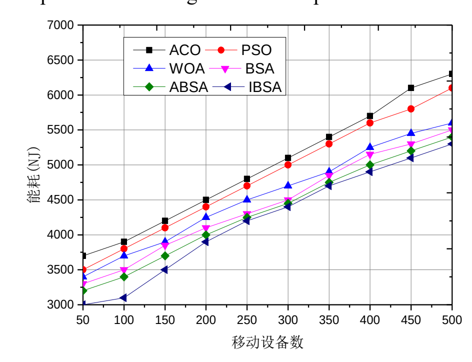


图 7　任务数量 1000 的六种算法设备能耗对比

Fig.7　Comparison of six devices' energy consumption with 1000 tasks

4　结束语

针对移动云计算下的任务调度存在完成时间长，移动设

备能耗高的问题，本文将移动云计算下的任务可行调度方案对应鸟群个体，以及调度目标函数作为鸟群算法适应度函数，通过对鸟群算法的感知系数和社会系数优化，在飞行行为中构建学习因子等措施提高了算法的性能，通过仿真实验说明本文算法在移动云计算的任务调度中能够有效的减少调度时间，降低能耗。下一步将研究成本因素在任务调度中的影响，优化目标调度模型，使之能够在移动云计算任务调度中发挥更好的作用。

参考文献：

[1] Chen shuang, Wang Yanzhi, Pedram M. A semi-markovian decision process based control method for offloading tasks from mobile devices to the cloud [C]// Pro of the 2013 Int Conf on Global Communications. Piscataway, NJ: IEEE, 2013: 2885-2890

[2] Deng Shuiguang, Huang Longtao, Taheri J, *et al.* Computation offloading for service workflow in mobile cloud computing [J]. IEEE Trans on Parallel and Distributed Systems, 2015, 26 (2): 3317-3329

[3] Khan A, Ahirwar K. Mobile cloud computing as a future of mobile multimedia database [J]. International Journal of Computer Science and Communication, 2011, 2 (1): 219-221

[4] Miettien A P, Nurminen J K. Energy of mobile clients in cloud computing [C]// Proc of the USENIX Int Conf on Hot Topcis in Cloud Computing. Berkeley, CA: USENIX Association, 2010: 14-21

[5] Balamurugan M, Akila V. Effective processor selection on heterogeneous computing [C]// Proc of the 2016 Int Conf on Science Technology Engineering and Management Piscataway, NJ: IEEE, 2016: 13-16

[6] Feng Bailong, Gao Jing. Distributed parallel Needleman Wunsch algorithm on heterogeneous cluster sysytem [C]// Proc of the 2015 Int Conf on Netowork and Information Systems for Computers. Piscataway, NJ: IEEE, 2015: 12-18

[7] Ra M, Sheth A. Mummert L, *et al.* Odessa: Enabling interactive perception applications on Mobile Systems, Applications, and Services. New York: ACM, 2011: 43-56

[8] Terzopoulos G, Karatza H. Dynamic voltage scaling scheduling on power-aware clusters under power constraints [C]// Proc of the 17th Int Conf on Distributed Simulation and Real Time Applications. New Yok: ACM, 2013: 72-78

[9] Saravanan S, Venkatachalam V. Advance MapReduce task scheduling algorithm using mobile cloud multimedia services architecture [C]// Proc of the 6 th Int Conf on Advanced Computing. Piscataway, NJ: IEEE, 2014: 21-25

[10] Deshmukh N, Deorankar A. Minimizing energy consumption in transmission efficient wireless sensor network [C]// Proc of the Int Conf on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics Piscataway, NJ: IEEE, 2016: 475-479

[11] 宋祖尧，戴月明. 移动云计算环境下多工作流任务调度的联合优化方法 [J/OL]. 计算机应用研究: 1-7 [2020-02-20]. https://doi. org/10. 19734/j. issn. 1001-3695. 2019. 02. 0058. (Song zu-yao, Dai yue-ming. Joint optimization method for multi-workflow task scheduling in mobile cloud computing environment [J/OL]. Application Research of Computers: 1-7 [2020-02-20]. https://doi. org/10. 19734/j. issn. 1001-3695. 2019. 02. 0058.)

[12] 胡海洋，刘润华，胡华. 移动云计算环境下任务调度的多目标优化方法 [J]. 计算机研究与发展, 2017, 54 (9): 1909-1919. (Hu hai-yang, Liu run-hua and Hu hua. Multi-Objective Optimization for Task Scheduling in Mobile Cloud Computing [J]. Journal of Computer Research and Development, 2017, 54 (9): 1909-1919)

[13] Lin X, Wang Y, Xie Q, *et al.* Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment [J]. IEEE Transactions on Services Computing, 2014, 8 (2): 175-186.

[14] Peng H, Wen W S, Tseng M L, *et al.* Joint optimization method for task scheduling time and energy consumption in mobile cloud computing environment [J]. Applied Soft Computing, 2019, 80: 534-545.

[15] Wang T, Wei X, Tang C, *et al.* Efficient multi-tasks scheduling algorithm in mobile cloud computing with time constraints [J]. Peer-to-Peer Networking and Applications, 2018, 11 (4): 793-807.

[16] Tang C, Hao M, Wei X, *et al.* Energy-aware task scheduling in mobile cloud computing [J]. Distributed and Parallel Databases, 2018, 36 (3): 529-553.

[17] 王祝琳. 移动云计算中任务分配策略研究 [D]. 长春: 吉林大学. 2016: 34-38. (Wang zhu-lin. Research on Task Allocation Strategy in Mobile Cloud Computing [D]. Chang chun. Lijin University. 2016: 34-38.)

[18] Shye A, Scholbrock B, Memik G Into the wild: studying real user activity patterns to guide power optimizations for mobile architectures [C]. IEEE/ACM International Symposium on Microarchitecture. ACM, 2009: 168-178.

[19] Kansal A, Zhao F, Liu J, *et al.* Virtual machine power metering and provisioning [C]. Proceedings of the ACM International Symposium on Computer Architecture. 2010: 326-337.

[20] 李建州. 安卓智能手机功耗管理评测机制的研究 [D]. 西安: 西安电子科技大学, 2014: 36-37. (Li jian-zhou. Research on the Android Power Management and Evalutaion Mechanism [D]. Xi an: XiDian university. 2014: 36-37.)

[21] Meng X B, Gao X Z, Lu L, *et al.* A new bio-inspired optimisation algorithm; Bird Swarm Algorithm [J]. Journal of Experimental & Theoretical Artificial lntclligcncc. 2016, 28 (4): 673-687.

[22] 樊晓红，万仁霞. 基于聚集度改进的多过程搜索鸟群算法 [J]. 科学技术与工程, 2019, 19 (16): 180-186. (Fan xiao-hong Wan ren-xia. An Improved Bird Swarm Algorithm of Multi-process Searching Based on Aggregation [J] , Science Technology and Engineering. 2019, 19 (16): 180-186.)

[23] 李延延，万仁霞. 一种改进的鸟群算法 [J]. 微电子学与计算机, 2018, 35 (09): 79-84. (Li yan-yan, Wan ren-xia. An Improved Algorithm for Bird Swarm Optimization [J]. Microelectronics & Computer, 2018, 35 (09): 79-84.)

[24] 王建伟，彭亦功. 引入迁移和变异策略的改进鸟群算法及其在参数估计中的应用 [J]. 华东理工大学学报（自然科学版）, 2018, 44 (04): 617-624. (Wang jian-wei, Peng yi-gong. Improved Bird Swarm Algorithm Based on Migration and Mutation Strategy and Its Application in Parameter Estimation [J]. Journal of East China University of Science and Technology, 2018, 44 (04): 617-624.)